

# 2011. május 19.

---

## - 19. oldal:

int, char és decimal típusok hosszának illetve értéktartományának javítása

## - 35. oldal:

Régi szöveg:

Az operátor úgy működik, hogy a kérdőjel előtti kifejezést kiértékeli, majd megnézi, hogy a kifejezés igaz vagy hamis. **Ha igaz, akkor az egyenlőségjel utáni** érték lesz a teljes kifejezésünk értéke, ha pedig hamis, akkor pedig a kettőspont utáni.

Javított szöveg:

Az operátor úgy működik, hogy a kérdőjel előtti kifejezést kiértékeli, majd megnézi, hogy a kifejezés igaz vagy hamis. **Ha igaz, akkor a kérdőjel utáni** érték lesz a teljes kifejezésünk értéke, ha pedig hamis, akkor pedig a kettőspont utáni.

## - 46. Oldal:

Régi szöveg:

Mivel a ciklusfeltétel után blokkot nyitunk, azt hinné az ember, hogy a ciklusváltozó a lokális lesz a ciklus blokkjára nézve, de ez nem fedi a valóságot. A ciklusfeltételen belül deklarált ciklusváltozó lokális lesz a *ciklust tartalmazó blokkra* nézve.

Javított szöveg:

Mivel a ciklusfeltétel után blokkot nyitunk, azt hinné az ember, hogy a ciklusváltozó a lokális lesz a ciklus blokkjára **(a for után következő kapcsos zárójelekkel határolt részre)** nézve, de ez nem fedi a valóságot. A ciklusfeltételen belül deklarált ciklusváltozó lokális lesz a *ciklust tartalmazó blokkra* **(vagyis ebben az esetben a teljes Main függvényre)** nézve.

## - 82. Oldal

Régi szöveg:

Private: csakis az osztályon belül elérhető, **illetve a leszármazott osztályok is láthatják, de nem módosíthatják** (a származtatás és öröklődés hamarosan jön) (UML jelölés: -).

Javított szöveg:

Private: csakis az osztályon belül elérhető, **a leszármazott osztályok nem láthatják és nem is módosíthatják** (a származtatás és öröklődés hamarosan jön) (UML jelölés: -).

## - 87. Oldal

Régi szöveg:

Egy osztály példányosításához legalább egy *public* elérhetőségű konstruktorra van szükség, egyébként nem fordul le a program.

Javított szöveg:

Egy osztály példányosításához a példányosítást végző programrész számára látható kell legyen a példányosítandó osztály konstruktora.

## - 152. Oldal

Régi szöveg:

A probléma gyökere, hogy postfixes operátort egyszerűen nem készíthetünk (ezt egyébként maga a Microsoft sem ajánlja) (illetve készíthetünk, de nem fog úgy működni, ahogyan szeretnénk).

Javított szöveg:

A probléma, hogy kifejezetten postfixes operátort nem tudunk definiálni, az általunk készített kód minden esetben a prefix operátort jelenti. Viszont meghívhatjuk az általunk definiált operátorokat postfixes alakban, ekkor az adott objektumot eltárolja a rendszer, meghívja az értéket növelő kódrészletet, és visszadja az első lépésben félrerakott értéket. Ugye ez értéktípusok esetében tökéletesen működik, de referenciatípusoknál az első lépésben nem az értéket hanem a referenciát tároljuk, vagyis a későbbi változás itt is érvényben lesz.

Ugyanakkor létezik módszer arra, hogy mégis megoldjuk a fenti problémát: az operátornak egy teljesen új objektumot kell visszaadnia, ekkor erre már nem mutat korábbi referencia, vagyis

biztonságosan használható (viszont az új művelet (az objektum létrehozása) miatt a teljesítményre negatív hatással lehet (ez persze néha elfogadható)).

Megjegyzés: az ezutáni oldalak eltolódnak az eredetihez képest (a tartalomjegyzék természetesen frissítve), az ezután következő változások az új oldalszámokat használják.

## - 164. Oldal

Régi szöveg:

A delegate hívásakor az összes a listáján lévő metódust meghívja. A delegate-ekhez a += illetve + operátorokkal hozzáadni a -= és – operátorokkal elvenni tudunk metódusokat:

Javított szöveg:

A delegatekhez egynél több metódust is hozzáadhatunk a += és + operátorokkal, valamint elvehetjük őket a -= és – operátorokkal. A delegate hívásakor az összes a listáján lévő metódust meghívja a megadott paraméterre.

## - 195. Oldal

Régi szöveg:

A memóriacím a memória egy adott byte-jára mutat (**vagyis a pointer növelése/csökkentése egy byte-al rakja odébb a mutatót**), amely az adott objektum kezdőcíme.

Javított szöveg:

A memóriacím a memória egy adott byte-jára mutat (**vagyis a pointer növelése/csökkentése a pointer típusának megfelelő mennyiségű byte-al rakja odébb a mutatót, tehát egy int pointer esetén ez négy byte lesz**), amely az adott objektum kezdőcíme.